A Large Ion Collider Experiment

European Organisation for Nuclear Research

The ALICE Storage System:

an Analysis of the Impact on the Performance of the Configuration Parameters and of the Load of Concurrent Streams

Adriana Telesca

PH/AID CERN

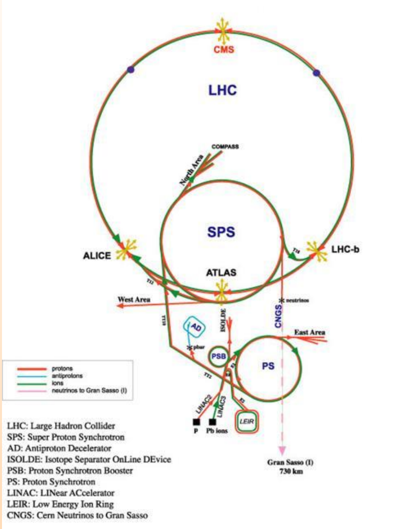
IEEE-NPSS Technical committee on Computer Applications in Nuclear and Plasma Sciences
May 10-15, 2009 IHEP Beijing

This research project has been supported by a Marie Curie Initial Training Network Fellowship of the European Community's Seventh Framework Programme under contract number (PITN-GA-2008-211801-ACEOLE)

Outline

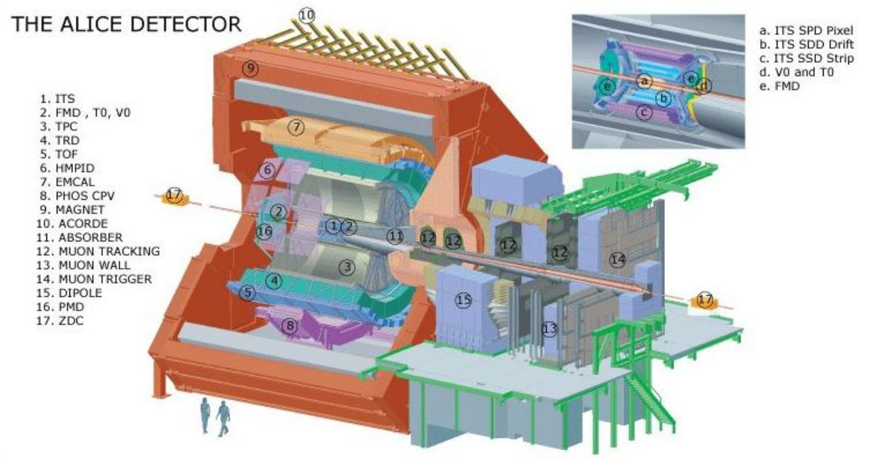
1. Introduction
   - The ALICE experiment
2. Hardware and benchmarking software performance
   - Hardware performance test
   - Performance tests with single stream
   - Performance tests with multiple streams
3. Hardware and application software performance
   - Performance with DATE
   - Nehalem architecture: performance
   - Data compression
   - Performance improvement
4. Conclusion

CERN, LHC and ALICE



LHC: Large Hadron Collider
SPS: Super Proton Synchrotron
AD: Antiproton Decelerator
ISOLDE: Isotope Separator OnLine DEvice
PSB: Proton Synchrotron Booster
PS: Proton Synchrotron
LINAC: LINear ACcelerator
LEIR: Low Energy Ion Ring
CNGS: Cern Neutrinos to Gran Sasso

ALICE detectors



Performance requirements

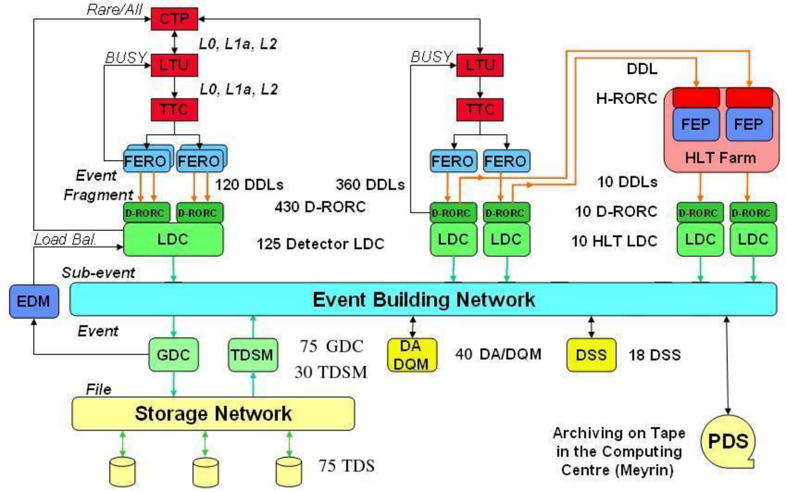
The DAQ system has the following requirements :

- an aggregate event building bandwidth of up to 2,5 GBytes/s
- a storage capability of up to 1,25 GBytes/s

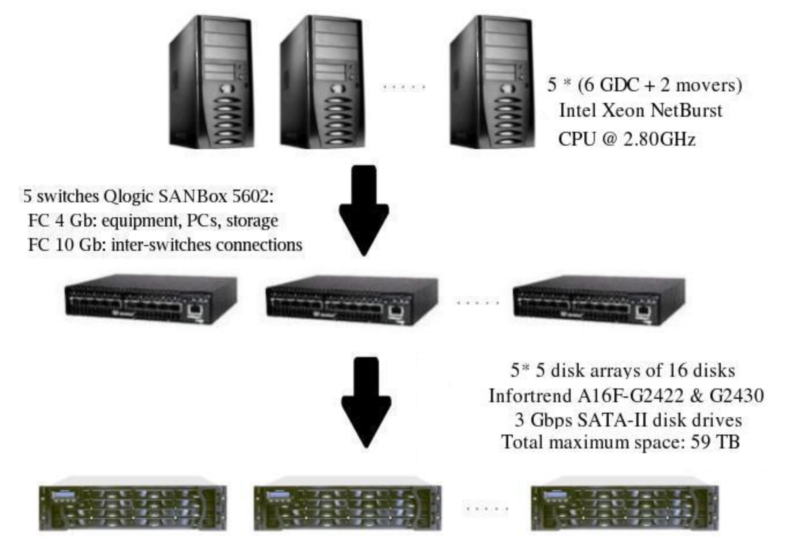
which result in a total of more than 1 PBytes of data every year.

This makes the performance of the mass storage devices a dominant factor for the overall system behavior and throughput.

Trigger - DAQ - HLT ’09



Current storage system

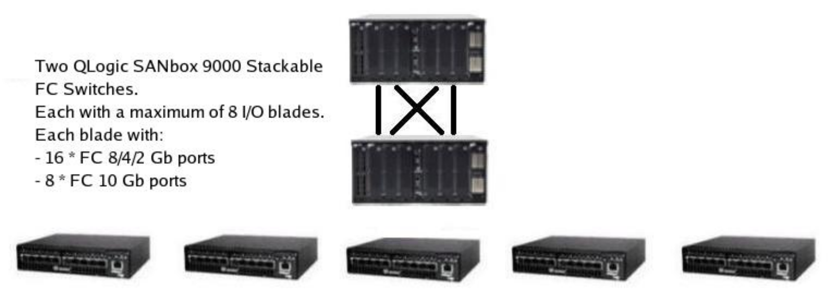


5 * (6 GDC + 2 movers)
Intel Xeon NetBurst
CPU @ 2.80GHz

5 switches Qlogic SANBox 5602:
FC 4 Gb: equipment, PCs, storage
FC 10 Gb: inter-switches connections

5* 5 disk arrays of 16 disks
Infortrend A16F-G2422 & G2430
3 Gbps SATA-II disk drives
Total maximum space: 59 TB

Deployment of the ALICE Storage Area Network in ’09



Two QLogic SANbox 9000 Stackable FC Switches.

Each with a maximum of 8 I/O blades.

Each blade with:

- 16 * FC 8/4/2 Gb ports
- 8 * FC 10 Gb ports

Test storage parameters and test software

Storage configuration parameters which can impact the system performance are:

- Block size
- File size
- RAID configuration
- Storage array configuration

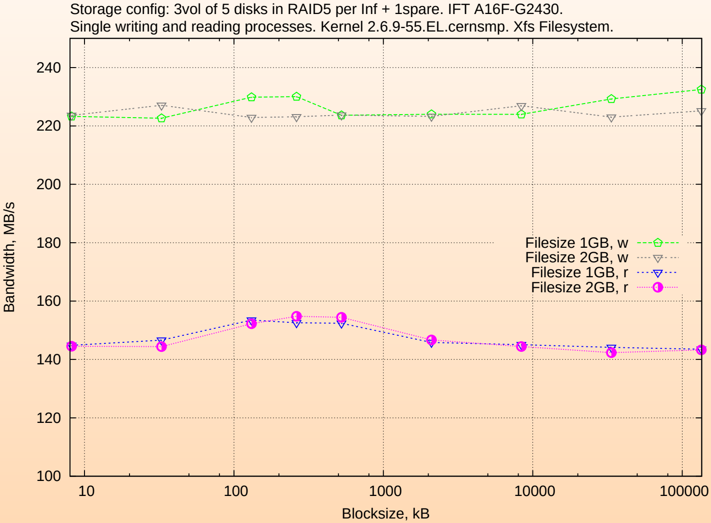
The test software is a standalone client called lmdd which:

- copies a specified input file filled by random data to a specified output;
- can be run simultaneously with other lmdds to perform parallel writing/reading streams;
- prints out the timing statistics.

All tests have been done with the hardware in production.

Block size and File size

Rate performance according to different file and block sizes. Xfs Unix file system.



Storage setup parameters

Storage configuration parameters chosen are:

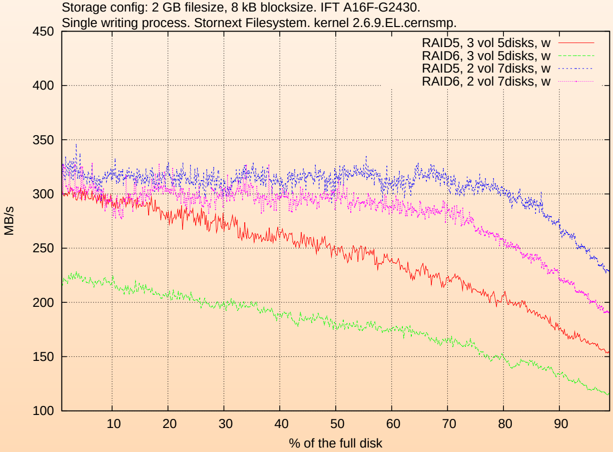
- Block size 8 KB
- File size 2 GB
- StorNext 3.1.2 cluster file system

Storage configuration parameters tested are:

- RAID configuration :
  - RAID 5
  - RAID 6
- Storage array configuration (16 disks):
  - 3 volumes of 5 disks + 1 spare
  - 2 volumes of 7 disks + 2 spares

Single writing

Storage performance tested according to the volumes/RAID configuration by performing single writing and reading operations from one GDC to one disk volume.

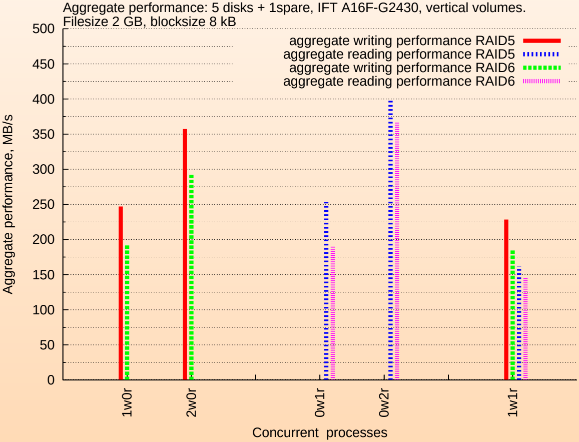


Multiple concurrent operations

- Concurrent activities on the same volume produce disastrous results
- The StorNext cluster file system allows to define an "affinity" which associates a file system folder to a logical unit. In this way we can address concurrent streams to different volumes
- An investigation on the coexistence of more streams on the same disk array is needed

Multiple concurrent writings and readings

Storage performance tested according to RAID configuration by performing concurrent writing and reading operations from two GDCs to two disk volumes.



DATE

DATE (ALICE Data Acquisition and Test Environment) is the software framework of the ALICE DAQ.

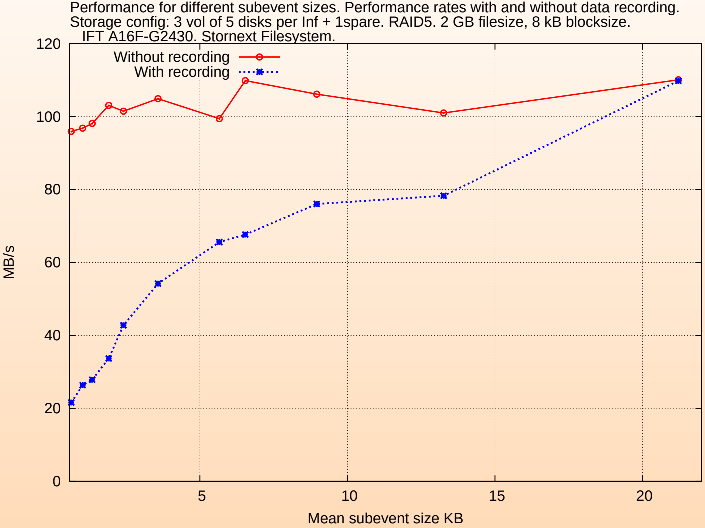
The DATE system performs different functions:

- Readout
- Event building
- Data recording
- Data formatting with ROOT format

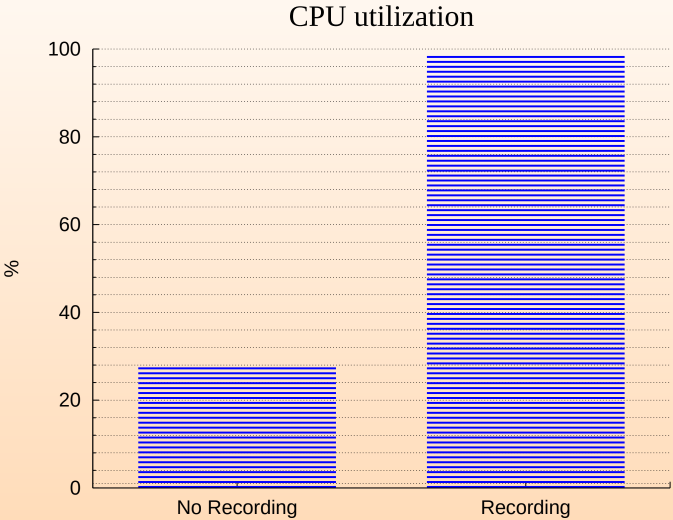
Parameters which can impact the system performance are:

- Mean subevent size
- Number of streams writing data

Performance for different subevent sizes

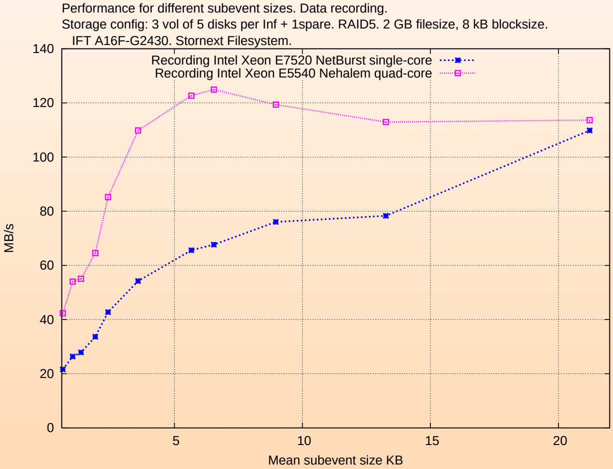


CPU utilization



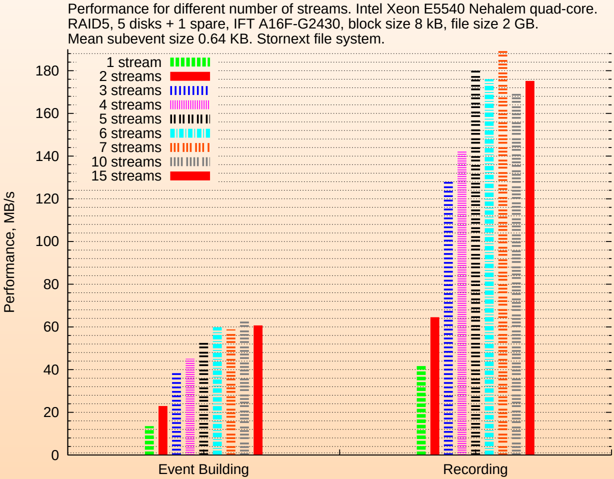
Recording performance with Nehalem architecture

We compared the data recording performance of the current PCs used in the ALICE deployment and the new generation of HP PCs based on the Nehalem microarchitecture.



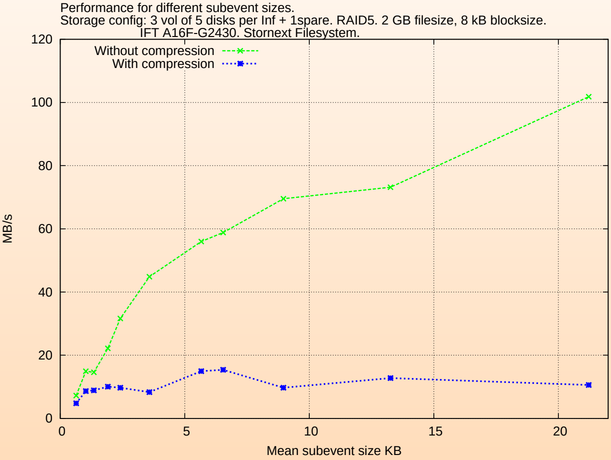
Recording performance with more streams

By exploiting the multi core and hypertrading functionalities, we see improved performance.



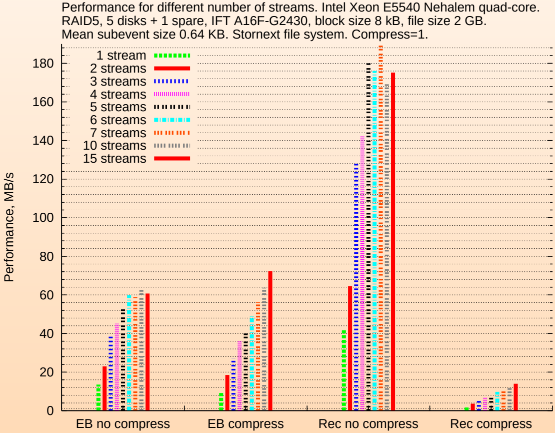
Performance with compression

We see the performance when compression is enabled.



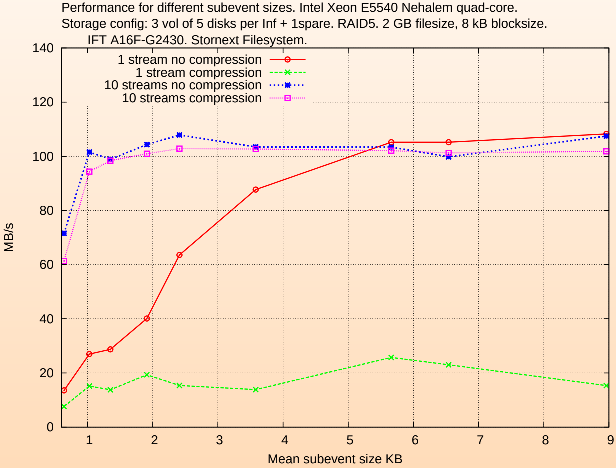
Performance with and without compression with multiple streams

We see the event building and data recording rates when compression is enabled with multiple streams.



Data compression with multiple streams

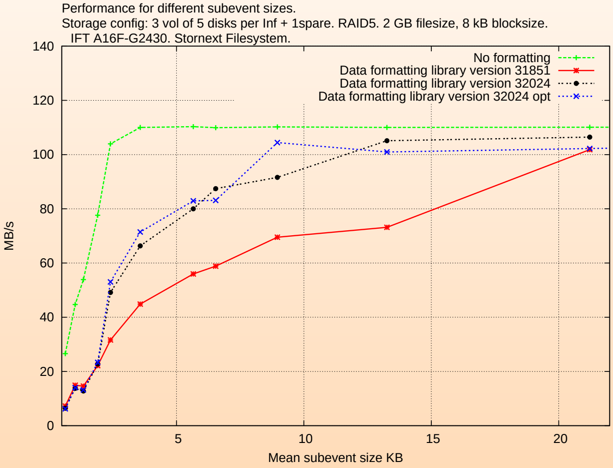
We see the bandwidth rate with and without compression with multiple streams.



Performance for different subevent sizes. Intel Xeon E5540 Nehalem quad-core.
Storage config: 3 vol of 5 disks per Inf + 1spare. RAID5. 2 GB filesize, 8 kB blocksize.
IFT A16F-G2430. Stornext Filesystem.

Event building for different data formatting library versions

We see the performance rate for different versions of the data formatting library.



Conclusion

- Storage hardware and system software provide adequate performance for ALICE.
- The storage hardware provides different performance for different configurations.
- The user CPU utilization impacts the software and hardware performance. We can obtain performance compatible to the ALICE needs for subevent sizes bigger than 20 kB.
- Tests on the new generation of HP machines based on the Nehalem microarchitecture, demonstrated that one core of this new architecture can improve the performance by up to 100%.
- By exploiting the multi-core functionality we can improve the performance. We have to optimize the number of streams according to the number of cores.
- Development is in progress to improve the performance and to reduce the data volume overhead.